

# Detecting Volatility Shift in Data Streams

## –Supplementary–

---

**Algorithm 1: SEED Algorithm**


---

```

1 Initialize window  $W$  as a list of blocks  $\{B_0, \dots, B_t\}$ 
  each with size of  $n$  ;
2 Boolean:  $hasDrift \leftarrow false$  ;
3 Integer:  $compressCount \leftarrow 0$  ;
4 for each  $t > 0$  do
5    $setInput(x_t, W)$ ;
6   return  $hasDrift$ 
7 end
8 Function  $setInput$  (item  $k$ , List  $W$ )
9    $addElement(k, W)$ ;
10  for every split of  $W$  into  $W = W_L.W_R$  do
11    if  $|\mu_{w_L} - \mu_{w_R}| > \epsilon_{cut}$  then
12       $hasDrift \leftarrow true$ ;
13      remove all blocks in  $W_L$ ;
14    end
15  end
16 end
17 Function  $addElement$  (item  $k$ , List  $W$ )
18  if Block at tail of  $W$  is full then
19    create a new Block  $B$  with content  $k$ ;
20     $W \leftarrow W \cup \{B\}$  (add  $B$  to tail of  $W$ );
21     $compressionCheck(W)$ ;
22  else
23    add  $k$  into tail block of  $W$ 
24  end
25 end
26 end
27 Function  $compressionCheck$  (List  $W$ )
28   $compressCount++$ ;
29  if  $compressCount = compressionInterval$  then
30    for each two consecutive block  $B_t$  and  $B_{t+1}$ 
31      do
32        if  $|\mu_{B_t} - \mu_{B_{t+1}}| < \epsilon'$  then
33           $B_t \leftarrow merge(B_t, B_{t+1})$ ;
34        end
35      end
36 end

```

---



---

**Algorithm 2: Volatility Detector**


---

```

1 Initialize Buffer  $B$  and Reservoir  $R$ ;
2 Boolean:  $volatilityShift \leftarrow false$ ;
3 for each  $t > 0$  do
4    $j \leftarrow addToBuffer(x_t, B)$ ;
5    $addToReservoir(j, R)$ ;
6    $RelativeVariance \leftarrow \frac{\sigma_B}{\sigma_R}$ ;
7   if  $RelativeVariance \leq 1.0 \pm \beta$  then
8      $volatilityShift \leftarrow true$ ;
9   end
10 end
11 Function  $addToBuffer$ (item  $k$ , Buffer  $B$ )
12   add  $k$  as tail of  $B$ ;
13   return head of  $B$ ;
14 end
15 Function  $addToReservoir$ (item  $k$ , Reservoir  $R$ )
16    $rPos \leftarrow random()$ ;
17    $R[rPos] \leftarrow k$ ;
18 end

```

---